



Memoria Sistemas Informáticos

Videojuego 3D para el estudio de
procesos cognitivos

Sergio Primo Galán

Proyecto de Sistemas Informáticos, Facultad de Informática, Universidad Complutense
de Madrid

Titulación: Ingeniería informática superior
Curso académico 2014-2015

Director en la Université de Montréal: Claude Frasson
Director en la Universidad Complutense de Madrid : Baltasar Fernández Manjón

Autorización de difusión y utilización

El autor abajo firmante autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor, tanto la propia memoria, como el código, los contenidos audiovisuales incluso si incluyen imágenes del autor, la documentación y/o el prototipo desarrollado.

El autor,

Sergio Primo Galán

Agradecimientos

Gracias al laboratorio Heron-JVI de la universidad de Montréal por acogerme el segundo semestre.

En especial a Claude Frasson, Asma Ben Kheder y Bel Lamine Sahbi por haberme ayudado a desarrollar este proyecto. También he de agradecer a Sébastien Oullet por darme un pequeño empujón al principio con Unity 3d y ayudarme con su código libre.

A Baltasar Fernández Manjón por ayudarme a corregir y mejorar ciertos aspectos del proyecto.

Y por último a todo el mundo, sobre todo mi novia y los amigos de Montréal, que estuvieron ahí en todo momento apoyándome para seguir trabajando y sacar adelante el proyecto.

Índice

1. Introducción.....	10
2. La salud en los videojuegos	12
2.1 ¿Cómo nos beneficia en la salud un videojuego?	12
2.1.1 Stress	12
2.1.2 Terapéuticos.....	15
2.1.3 Mejora de las habilidades motoras de los niños preescolares	16
2.1.4 Mitigación del dolor	16
2.1.5 Mejora de la visión	17
2.1.6 Mejora en la toma de decisiones	18
2.1.7 Cognitivos	18
2.2 Cognición.....	19
2.2.1 ¿Qué es la cognición?.....	19
2.2.2 Tipos de cognición.....	21
2.2.3 Los videojuegos en la cognición	23
3. Estructura del proyecto	25
3.1 Objetivos generales.....	25
3.2 Metodologías	26
3.3 Tecnologías.....	27
3.4 Planificación	28
4. Arquitectura.....	30
4.1 Estructura del videojuego	31
4.2 Modelado 3D.....	34
4.2.1 Paquetes comprados a través de la asset store de Unity 3d	34
4.2.2 Objetos añadidos o modificados por Blender.....	36
4.3 Scripts.....	37
4.4 Escenas.....	41
4.5 Animaciones	43
4.6 Audio	45
4.7 GUI.....	46

5. Pruebas	48
5.1 Entorno 3d.....	48
5.2 Mini juego Lógico	49
5.3 Mini juego de diagnósticos	49
5.4 Pruebas generales	50
6. Manual de usuario	51
6.1 Controles	51
6.2 Menú de pausa.....	52
6.3 Mini juegos	53
7. Conclusión y trabajo futuro	54
7.1 Conclusiones y principales aportaciones	54
7.2 Trabajo futuro	55
8. Glosario.....	57
9. Bibliografía	59

Índice de ilustraciones

Ilustración 1. Efectos de los videojuegos violentos y no violentos en el la coherencia cardiaca al principio y durante el juego.....	14
Ilustración 2. Halo 2.....	19
Ilustración 3. Procesos cognitivos.....	20
Ilustración 4. Diagrama de Gantt.	29
Ilustración 5. Vistas de Unity 3d.....	31
Ilustración 6. Herencia en Unity 3d.	33
Ilustración 7. Diagrama de flujo de las escenas.	42
Ilustración 8. Animación en Unity 3d.	44
Ilustración 9. Mini juego diagnóstico.....	47
Ilustración 10. Interacción con objetos.	52
Ilustración 11. Menú de pausa.....	52
Ilustración 12. Menú para el control del volumen de la música.....	53

Abstract

This Project was born in the laboratory Heron, at the University of Montréal, Canada.

The idea was learn and develop a game since the beginning, for a specific purpose, test and study the cognition and the patient's reasoning while he is playing the videogame.

Meetings were held periodically with the people involved to achieve these objectives. Having knowledge about cognition and the skills they wanted to study, they guided me in the development.

A 3d environment was built to immerse the player in the videogame, with some cognitive exercises. It wasn't enough to test various cognitive skills, so few mini games were created, each one searching a specific skill to test.

The game takes place in a hospital, the patient is amnesic and he needs to tests his cognitive skills.

The development of this game, Hospital Amnesia, was difficult at the beginning, because of my lack of knowledge of the engine, but I learned fast and I applied what I learned to the game.

The game engine used was Unity 3d, a free software with a big potential for independent developer. Despite having the pro version in the laboratory, it was not necessary, and the free version offers the possibility to continue working on my return, and correct or expand what the laboratory needs.

Keywords:

- Video game
- Unity 3d
- Hospital Amnesia
- Health
- Cognition
- Cognitive
- Blender

Resumen

Este Proyecto nació en el laboratorio Heron, en la Universidad de Montréal, Canadá.

La idea era aprender y desarrollar un videojuego desde el principio, para un propósito específico, examinar y estudiar la cognición y el razonamiento del paciente mientras juega al videojuego.

Reuniones se llevaron a cabo periódicamente con las personas involucradas para alcanzar estos objetivos. Teniendo conocimientos sobre la cognición y de las habilidades que querían estudiar, ellos me guiaban en el desarrollo.

Un entorno 3d fue construido para sumergir al jugador en el videojuego, con algunos ejercicios cognitivos. No era suficiente para probar varias habilidades cognitivas, así que varios mini juegos se crearon, cada uno buscando una habilidad específica a examinar.

El juego se ubica en un hospital, el paciente es amnésico y tiene que poner a prueba sus habilidades cognitivas.

El desarrollo de este juego, Hospital Amnesia, fue complicado al principio, a causa de mi desconocimiento del motor, pero aprendí rápido y apliqué lo aprendido al juego.

El motor de juego utilizado fue Unity 3d, un software gratis con un gran potencial para desarrolladores independientes. A pesar de disponer de la versión pro en el laboratorio, no era necesaria, y encima la versión gratis ofrecía la posibilidad de seguir trabajando a mi vuelta, y así corregir o ampliar lo que el laboratorio necesitará.

Palabras clave:

- Videojuego
- Unity 3d
- Hospital Amnesia
- Salud
- Cognición
- Cognitivo
- Blender

Introducción

En este proyecto fin carrera quería desarrollar un videojuego enfocado al estudio de las funciones cognitivas del ser humano. El juego se ha desarrollado en exclusiva y bajo petición del laboratorio Heron de la universidad de Montréal, Montréal, Canadá. En las reuniones con el director del laboratorio se decidían qué cambios realizar. Por tanto el diseño del juego se realizó en función de los requisitos de la entidad académica, y no sólo míos. Algunos aspectos del juego podrían haberse realizado de otra forma.

Las pruebas en pacientes se realizarán a partir de septiembre 2015.

Es difícil saber cuándo nacieron los videojuegos, la fecha exacta se desconoce, pero los primeros indicios datan al finalizar la segunda guerra mundial, cuando el ocio vuelve a ser una prioridad en la vida de la población. Los primeros videojuegos que se desarrollaron fueron un ajedrez o el conocido cruz en raya.

No es hasta la década de los 60, que se empiezan a desarrollar los videojuegos modernos en Estados Unidos, y dónde empieza todo. El boom estalla a principios de los 80, con la comercialización masiva de estos juegos electrónicos.

Pero no sólo se pensó en el entretenimiento sin ningún otro fin. En la misma época se empiezan a desarrollar juegos educativos. Juegos tales como *Lemonade Stand* (en 1979 fue llevado a la plataforma Apple II), que consistía en un puesto de limonada, en el cual el jugador tenía que elegir cuantos limones comprar, cómo hacer publicidad y a qué precio vender la limonada. El juego enseñaba complejas lecciones sobre negocios y economía y fue uno de los primeros en usar este método para ello. Varios juegos como *Lemonade Empire*, *Lemonade Tycoon* o *Hot Dog Stand* se inspiraron de este.

Snooper Troops (1982) se trató del primer juego en tener que usar el razonamiento, recopilando pistas y resolviendo el caso a partir de estas. El juego desarrollado en este proyecto tendrá una parte basada en este concepto.

Hasta 1986 no se creó uno de los juegos educativos más famosos y utilizados tanto en colegios como en casas, se trata de *Reader Rabbit*. El primero de la serie enseñaba la lengua a los más pequeños, tales como la lectura y ortografía básica. Más adelante se iría desarrollando para enseñar a alumnos algo más mayores, y otros campos como las matemáticas.

Como vemos, los juegos no tienen que servir solo a entretenernos, también puede enseñarnos de forma divertida en muchos campos, lenguas, matemáticas, razonamiento, aviación, medicina, etc...

Desde hace unos años, se están integrando en el campo de la medicina. Los investigadores afirman que tienen efectos beneficiosos para la salud. Se usan como herramienta de fisioterapia y terapia ocupacional en algunos pacientes puesto que es fácil de integrar. (Smith & Schoene, 2012) Se utilizan sobre todo para que los ejercicios que pueden resultar bastante repetitivos algunas veces, se hagan más a menos gracias a esta integración. La plataforma Wii fue pionera en este ámbito con algunos juegos como *wii fit* (Xian Ying, Kakinami Lisa, Peterson Eric D., Mustian Karen M., 2014) , estrenado en 2007, que incorpora el innovador control de la consola y puntuaciones, ayudando a la constancia y promoviendo la superación personal. Uno de los aspectos de los videojuegos a parte del entretenimiento obviamente, es la competencia, los jugadores luchan por ser el mejor entre todos. Se le llama *Gamification*. Y esto se consigue con puntuaciones, aplicada en la medicina, podría conseguir que las personas se esfuercen más y así desarrollen aún más sus habilidades. En tres de los mini juegos de **Hospital Amnesia** se registrarán las puntuaciones para que el jugador se esfuerce totalmente.

Se han creado algunos juegos para probar las habilidades médicas de los jugadores, desde para todo el público, como sería *Trauma Center*, hasta profesionales. En 2011, *Seprtris* (Stephen, 2014) se creó con el propósito de enseñar a aspirantes a médicos. Con casos de estudio interactivos, el juego ayuda a los usuarios a aprender la mejor forma de tratar la sepsis, una complicada enfermedad que empieza como una infección bacteriana que puede progresar rápidamente a un tratamiento de por vida para la inflamación del cuerpo. El último mini juego de HA se parece a este juego, el jugador tendrá varios casos que tendrá que resolver, irán aumentando en dificultad.

La salud en los videojuegos

En el ámbito de los videojuegos se han realizado incontables juegos desde el primer 3 en raya hasta hoy en día. Desde los 8 bits hasta los gráficos en 4k, se han desarrollado juegos de todo tipo de géneros, todos ellos con el mismo propósito: entretener.

Pero no es el único objetivo, también se quiere enseñar a la gente a través de ellos, y estudiar el impacto de estos en cada ser humano.

2.1 ¿Cómo nos beneficia en la salud un videojuego?

2.1.1 Stress

Los beneficios son tanto mentales como físicos, dependiendo el juego en cuestión claro. Un estudio (Shankar, 2009) reveló que puede reducir el stress y la depresión, dando rienda suelta a la agresión y frustración. Se podría parecer a la famosa terapia que vemos en las películas, de destrozarse un vehículo con un martillo.

En este estudio, los investigadores de PopCap commissioned and funded se dieron cuenta de que los voluntarios que jugaban a Bejeweled (juego similar al famoso Candy Crush) habían mejorado su estado de ánimo y su ritmo cardiaco, comparado a los voluntarios que no jugaban. Uno de los pacientes indicó que la versión que más le gustaba era la cual no competía con nadie más, y por tanto podía concentrarse y relajarse al mismo tiempo.

Algunos juegos permiten que la gente alcance un ritmo en el que puede jugar sin esfuerzo ni aburrimiento. Al parecer, algunos juegos activan el sistema nervioso parasimpático, lo que puede reducir el aumento de tensión, lo cual es una respuesta natural.

Por tanto una de las razones por la cual pueden ser beneficiosos para la salud mental, es que en muchos países las personas no son capaces de desconectar, están siempre alerta y estresados. Por ello muchos necesitan juegos que produzcan una situación algo más estresante, pero si encuentran alguno que les suponga un desafío mental, les beneficiará como al resto, y entrar en un estado de inconsciencia, el cual tiene efectos saludables sobre el estrés y otros problemas mentales.

Compañías como Heartmath, of Boulder Creek, California, han desarrollado juegos en los cuales requieren que el jugador regule su ritmo cardiaco, obteniendo mejores respuestas en situaciones estresantes. Rollin McCraty, un psicólogo que trabaja en la empresa, afirmó que existían una docena de estudios demostrando que el ritmo cardiaco está relacionado con varios estados emocionales.

McCraty realizó un estudio en el que los soldados que habían aprendido a controlar su ritmo cardiaco jugando a algunos videojuegos eran más eficientes bajo presión, y que se recuperan más rápido de sucesos traumáticos.

Otros estudios realizados por funcionarios de prisión y enfermeras de algunos hospitales, demostraron que al otorgarles mayor control sobre sus estados, daban mejores resultados en el trabajo y reducía el riesgo de sobrecarga, tal y como McCraty creía.

Pero no todos los juegos aportan lo mismo, y es por eso que hay que saber elegir el juego para cada tipo de persona, dependiendo lo que uno quiera. Otro estudio relevó que los videojuegos agresivos pueden al contrario convertirnos mucho más agresivos (Hasan, Bègue, & Bushman, 2013). Tomando la misma referencia del ritmo cardiaco, demostraron que en cambio el ritmo cardiaco puede aumentar con juegos violentos.

Exactamente fue medido usando la coherencia cardiaca, que es la sincronización del ritmo de la respiración con el ritmo del corazón. Predijeron que la coherencia cardiaca podría mediar en la relación entre la exposición a los videojuegos violentos y la subsecuente agresión. Específicamente, que jugar a un videojuego violento disminuiría la coherencia cardíaca y que la coherencia cardíaca, a su vez, se correlaciona negativamente con agresión.

Se usó esta medida, puesto que ofrece más ventajas que a la simple medición del ritmo cardiaco. Está directamente relacionado a los efectos negativos como el stress a diferencia de otras medidas fisiológicas, puesto que es capaz de distinguir entre la regulación simpática y parasimpática del ritmo cardiaco. Es menos invasiva que otras medidas, que te obligan a llevar una pulsera, electrodos o similar para medir tu comportamiento fisiológico. La medición de la coherencia cardiaca es bastante estable, no reacciona ante los típicos movimientos musculares de un jugador, además de no

estar afectada por los self-report, ser bastante más barata de realizar y muy utilizada por los investigadores.

Se analizó con dos videojuegos, uno violento y otro no, en la fase inicial del videojuego y durante. Los resultados los podemos ver en la figura 1. Se ve claramente como con el juego no violento la coherencia cardiaca aumenta ligeramente, y con el violento esta disminuye casi un 50%.

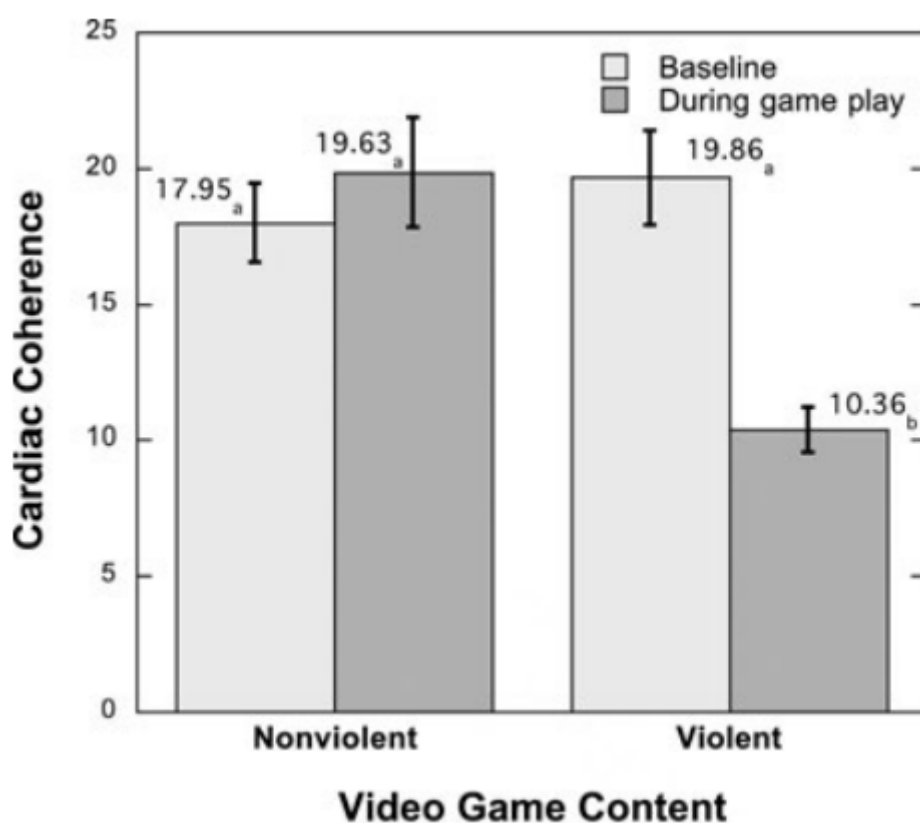


Ilustración 1 Efectos de los videojuegos violentos y no violentos en el la coherencia cardiaca al principio y durante el juego.

Sin embargo, los neurólogos afirman que también tienen efectos beneficiosos en el cerebro, como aumentar la función de ciertas regiones del encéfalo y mejorar el bienestar general. Más adelante lo veremos.

2.1.2 Terapéuticos

Una publicación del 2012 (Bulaj, 2012) publicada por investigadores de la Universidad de Utah en la revista Science Translational Medicine, afirmaba que los videojuegos pueden resultar terapéuticos y muestran beneficios para la salud.

Carol Bruggers es la autora principal del artículo “Patient-Empowerment Interactive Technologies”, una profesora de la universidad de Utah del departamento de pediatría y médico.

Bruggers mencionó: “Un creciente número de estudios publicados prevén cambios específicos relacionados con la salud del comportamiento y la autogestión de la obesidad, trastornos neurológicos, cáncer o asma. “

Los sanitarios también se beneficiarán de muchas oportunidades de usar juegos de video basados en incentivos en la gestión y prevención de enfermedades. Cada vez más empresas, organizaciones sin fines de lucro y centros académicos están involucrados en el diseño y la publicación de las tecnologías interactivas para enfermedades metabólicas, trastornos de salud mental, cáncer, derrame cerebral o rehabilitación.

Los investigadores de Utah dicen que los videojuegos pueden actuar como "intervenciones no farmacológicas [que] puede aumentar la capacidad de recuperación de los pacientes hacia diversos trastornos crónicos a través de los mecanismos neuronales que activan las emociones positivas y el sistema de recompensas."

"La gente juega juegos porque son atractivos. Ahora estamos empezando a entender cómo los juegos nos motivan, y cómo utilizar esta motivación para cambiar la atención de salud ", dice Altizer. "Si los juegos como el nuestro pueden ayudar a los pacientes a sentirse mejor y motivarlos a manejar su atención médica o terapia física, entonces creo que pronto veremos la comunidad médica diciendo: "¡Juego!"

Grzegorz Bulaj, profesor asociado de química medical en la Universidad de Utah, añade: "La investigación muestra que los videojuegos aumenta los niveles de dopamina en el cerebro, pero si las tecnologías interactivas pueden imitar las acciones farmacológicas de los medicamentos sigue siendo desconocido. Sin embargo, nuestro estudio apunta a que los videojuegos se convertirán en una parte de la medicina personalizada, ayudando y trayendo sonrisas a cada paciente, médicos, enfermeras y fisioterapeutas. Nuestro trabajo muestra que estos juegos son una gran promesa, pero

también que la entrega segura, eficaz y la diversión de los juegos terapéuticos son un desafío".

2.1.3 Mejora de las habilidades motoras de los niños preescolares

Investigadores de la Universidad de Deakin (Melbourne, Australia) llegaron a la conclusión de que los niños de preescolar que juegan videojuegos interactivos (similares a los de la Wii) tienen mejores habilidades motoras que aquellos que no juegan regularmente (Barnett, 2012). Estos investigadores y otro de la universidad de Wollongong realizaron un primer estudio con 53 niños entre 3 y 6 años para determinar si había alguna relación entre jugar a los videojuegos y los principales movimientos de habilidad de los niños.

Se repartió de 3 formas, exclusivamente juegos interactivos, no interactivos, o ambos. Entre 2 y 3 horas semanales jugaban.

El resultado del estudio demostró que aquellos que jugaron juegos interactivos desarrollaron más las habilidades que aquellos que lo hicieron con juegos no interactivos. El problema es que los investigadores no encontraron una relación entre las habilidades motoras y jugar a juegos interactivos o no.

2.1.4 Mitigación del dolor

Sam Brown, fue un militar estadounidense que sufrió graves heridas y quemaduras en Iraq, y tras muchos tratamientos en su país, le ofrecieron uno completamente distinto, SnowWorld ("Groundbreaking experiment in virtual reality uses video game to treat pain," 2012).

SnowWorld, es un pequeño juego de realidad virtual. En él, los jugadores pueden concentrarse en tirar bolas de nieve a los pingüinos u otros animales, escuchando música, y así olvidarse del dolor de la herida.

La realidad virtual de este juego fue ideada en la Universidad de Washington por dos psicólogos, Dr. David Patterson y el Dr. Hunter Hoffman. Esta realidad ofrece una distracción que oprime los sentidos y desvía la atención del cerebro de las señales de dolor.

Dr. Patterson ha trabajado con pacientes con graves quemaduras y explicó que la cura puede producir más dolor que la propia quemadura.

Patterson dijo: "No estás más que llevando a la persona a en un mundo alternativo. Y funciona tanto tiempo como la gente cree estar en el mundo virtual".

En 2011, el ejército llevó a cabo un pequeño estudio usando SnowWorld y obtuvo resultados impresionantes. Para los soldados con mayor dolor, SnowWorld funcionó mejor que la morfina.

“Siento mucho menos dolor, a lo que experimenté anteriormente” fue una de las frases que confesó el militar Brown.

2.1.5 Mejora de la visión

Doctora Maurer, directora del Laboratorio de Desarrollo Visual de la Universidad McMaster en Ontario, es autora, junto a su marido, Charles, del pionero libro de 1988 "El Mundo del recién nacido", un inventario de lo que perciben los bebés y experiencia. En los últimos años ha estado dirigiendo un estudio de seguimiento de los bebés que nacen con deficiencias visuales en la vida posterior (Jeon, Maurer, & Lewis, 2012). Este estudio longitudinal es un intento de aprender principios de privación sensorial que afecta a la visión el resto de su vida. En especial ha realizado el seguimiento de los bebés con cataratas y otro grupo de jóvenes que había desarrollado cataratas a posteriori, a ambos se les había realizado cirugía.

Daphne Bavelier, de la Universidad de Rochester comenzó a publicar estudios (Green & Bavelier, 2006) que mostraban que los videojuegos mejoran la visión de las personas con visión normal. Si ellos ayudaron a los normalmente videntes, ¿por qué no a las personas con discapacidad? Había estudios donde los ambientes enriquecidos para ratas mejoraron su visión dañada y a La doctora Maurer se le ocurrieron los videojuegos.

Al final de un mes, probaron su visión de nuevo. Todos ellos mostraron una cierta mejora en un número de tareas. Algunos estaban viendo mejor con su mejor, peor o ambos ojos. La mayor mejora fue su capacidad de ver la dirección del movimiento. Podían ver los detalles más pequeños. También fueron capaces de ver las cosas que estaban en bajo contraste. Fue una señal de que el cerebro sensorial puede cambiar en los adultos.

Los juegos en primera persona de acción tienen todo lo necesario. Se necesita que el jugador controle todo el campo de visión, puesto que el enemigo puede venir de cualquier parte. El elevado ritmo provoca un subidón de adrenalina y de dopamina en el cerebro.

Al tener que tomar decisiones y estar atento a la pantalla, el jugador se sumerge en el juego. Esto es importante, puesto que otros investigadores probaron con otro juego, el Tetris, que no ofrecía las mismas condiciones y no obtuvieron ningún resultado relevante de los pacientes.

Una de las hipótesis es que estén cambiando el equilibrio de excitación e inhibición en las entradas a las neuronas en la corteza visual. Los videojuegos podrían estar formando nuevas conexiones, simplemente hacerlas resurgir o hacer más eficiente al cerebro para obtener respuestas de señales débiles.

2.1.6 Mejora en la toma de decisiones

Investigaciones anteriores mostraron que las personas que juegan videojuegos de acción tienen tiempos de reacción más rápidos que los que no juegan los juegos (Hsu, 2010). Los científicos hallaron que los jugadores son mejores en la toma de decisiones rápidas y precisas, basándose en la información que extraen de su entorno. Los investigadores usaron una serie de experimentos de decisión para compararlos (Granic, Lobel, & Engels, 2014). Los jugadores jugaban 5 horas semanales a juegos como el Grand Theft Auto: San Andreas o el Halo 2 (Ilustración 2).

Se les pidió que identificaran de qué manera se movían ciertos puntos a su alrededor, y se variaba el número de puntos que se movían en la misma dirección para aumentar o disminuir la dificultad del ejercicio. Los jugadores mostraron juicios más rápidos sin perder precisión.

Las personas que no solían jugar se vieron obligadas a jugar a videojuegos de acción, como "Call of Duty 2," durante 50 horas y mostraron una mejora en la toma de decisiones.

"Nuestros cerebros realizan constantemente inferencias probabilísticas – cuando circula a lo largo de una carretera, si detecta algún objeto en movimiento inesperado en el lado derecho de la carretera, una motocicleta por ejemplo, el cerebro calcula qué tan probable es que esté colisione con la moto, y entonces infiere en la probabilidad de si necesita dirigirse la izquierda o no ", explica Bavelier. "Este tipo de inferencia se utiliza cada vez que tomamos una decisión." Los videojuegos de acción dan una ventaja "al mejorar este proceso de inferencia", señaló, mientras que la estrategia o de rol juegos no tienen el mismo efecto.

Estos descubrimientos, entre otras cosas, podrían ayudar a mejorar la rehabilitación después de una lesión cerebral o ayudar a las personas con trastornos del aprendizaje.

2.1.7 Cognitivos

Este apartado lo trataremos más específicamente adelante, puesto que lo analizaremos en detalle.



Ilustración 2. Halo 2

2.2 Cognición

2.2.1 ¿Qué es la cognición?

La cognición es la facultad del ser viviente de procesar la información, analizarla y adquirirla a través de lo que llamamos conocimiento, mediante la percepción.

Se suele hablar más de ella como las capacidades mentales del individuo. Éstas pueden ser de muchos tipos tales como la atención, la memoria, el razonamiento lógico, la resolución de problemas o la velocidad de reacción, que serían los que hemos tratado en este trabajo, para estudiarlo en los pacientes.

Esta capacidad puede ser desarrollada, aunque también deteriorarse. Por ello buscamos si gracias a los videojuegos la cognición podría ser mantenida o incluso desarrollarse aún más para conservar o mejorar la salud de los pacientes que los utilicen.

Benjamin Bloom fue uno de los precursores en este campo, con su famosa Taxonomía de Benjamin Bloom” en la cual describía su teoría de las habilidades cognitivas las cuales dependen unas de otras. Las clasificó en 6 subáreas:

- **Conocimiento:** Implica el conocimiento de hechos específicos y conocimientos de formas y medios de tratar con estos. Así como los conocimientos de lo universal y de las abstracciones específicas de un determinado campo del saber. Suelen ser elementos a memorizar

- **Comprensión:** Es el aspecto más simple del entendimiento que consiste en captar el sentido directo de una comunicación o de un fenómeno, como la comprensión de una orden escrita u oral, o la percepción de algún hecho.
- **Aplicación:** El conocimiento de aplicación es el que concierne a la interrelación de principios y generalizaciones con casos particulares o prácticos.
- **Análisis:** El análisis implica la división de un todo en sus partes y la percepción del significado de las mismas en relación con el conjunto. El análisis comprende el análisis de elementos, de relaciones, etc.
- **Síntesis:** Concierne la comprobación de la unión de los elementos que forman un todo. Puede consistir en la producción de una comunicación, un plan de operaciones o la derivación de una serie de relaciones abstractas.
- **Evaluación:** Este tipo de conocimiento comprende una actitud crítica ante los hechos. La evaluación puede estar en relación con juicios relativos a la evidencia interna y con juicios relativos a la evidencia externa.

Estas áreas desde entonces han sido reestructuradas una vez teníamos más conocimiento de ellas, una reestructuración puede ser la de Anderson.

Las más conocidas e importantes son las que aparecen en la ilustración 3 de las cuales hablaremos más adelante.

PROCESOS COGNITIVOS

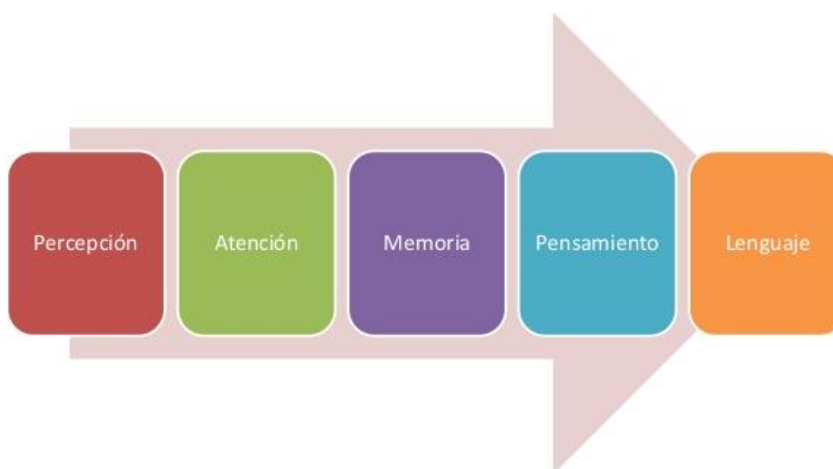


Ilustración 3. Procesos cognitivos

2.2.2 Tipos de cognición

Existen muchos más, a parte de los de Bloom y de los nombrados anteriormente, como el procesamiento de lenguaje o la toma de decisiones (aunque está incluida en otros ejercicios). Pero vamos a tratar los que aparecen en los mini juegos de HA.

Razonamiento

Se entiende razonamiento como la capacidad de resolver problemas, extraer conclusiones y aprender en consecuencia de ello, existiendo una conexión obviamente entre todo.

En el juego se ha realizado dentro del escenario 3D, una serie de ejercicios bastantes sencillos, para que el jugador no se desanime y dé rienda suelta a su razonamiento y su cerebro resuelva el problema.

El jugador tiene que resolver 2 combinaciones de puerta, cada una completamente independiente. En la primera, el paciente tiene que buscar dos números repartidos por la habitación. Para ello hay que reunir un par de piezas, de forma similar a miles de juegos de detectives, en los cuales tienes que resolver pistas.

La segunda, el jugador, al ver los colores del código y la pista, debería deducir de dónde sacar los números.

Atención

La atención es la capacidad que tiene alguien para entender las cosas o algo en concreto y darle un posible uso.

En el juego se usará una página web externa llamada Lumosity (<http://www.lumosity.com>), en la cual el jugador se registrará rápidamente y podrá realizar 3 mini juegos, en los cuales hay que utilizar la atención.

También en el razonamiento se necesita, es una capacidad que suele ir en conjunto con otras.

Memoria

La memoria es la capacidad de retener la información que nos rodea, de cualquier tipo, de textos a sensaciones u olores. En HA se necesita en 2 mini juegos y en la segunda combinación también habrá que retener algo de información.

El primero será también parte de Lumosity, que habrá que recordar una serie de patrones. El segundo será el típico juego de parejas con cartas, que habrá que ir emparejando las cartas con el mismo dibujo. La dificultad en ambos irá aumentando cada ronda.

Razonamiento lógico

Es el proceso mental que implica la aplicación de la lógica. A partir de la lógica se puede llegar a una conclusión. Se puede iniciar de varias formas, de observaciones (como en el juego de las combinaciones), hipótesis, o simplemente numéricos. En el mini juego de lógica de del juego, habrá una serie de preguntas numéricas, y otras con premisas, de las cuales habrá que deducir la respuesta. También se tendrá que resolver ciertos problemas.

En Lumosity también aparecerán en menor medida.

Pero dónde más se usará será en el último ejercicio, que con ayuda de ciertos conocimientos médicos, está enfocada al personal médico, para probar su razonamiento lógico y conocimientos. Habrá una serie de casos que irán apareciendo, y el jugador tendrá que resolverlos, encontrando la enfermedad que coincida, y seguidamente el o los tratamientos que se aplicará.

Existen dos tipos de razonamientos, el deductivo y el inductivo.

El razonamiento deductivo deduce una conclusión a partir de una serie de premisas, de lo general a lo particular.

El razonamiento inductivo consiste en a partir de hechos más individuales, como pueden ser experiencias, se extrae una conclusión general. Esto puede llevar a un fallo, que sea cierto es una probabilidad que depende del números de premisas, su características, etc...

Lenguaje

Se trata de la capacidad del ser humano para comunicarse vía escrita, oral u otra. Esta capacidad se trata ligeramente en el juego lógico con algunas preguntas pero no se hace mucho hincapié, es un proceso complicado de tratar, al haber varias lenguas y cada una ya de por sí tiene un vasto trabajo por delante, como puede ser el razonamiento textual.

2.2.3 Los videojuegos en la cognición

Jugar a estos juegos promueve una serie de habilidades cognitivas (Granic et al., 2014), especialmente en los juegos de acción en primera persona denominados shooter. Un estudio llegó a la conclusión de que las mejoras de habilidades espaciales en estos juegos se podrían asimilar a la de ciertos cursos universitarios que buscan mejorarlas. Encima estas habilidades, se pueden potenciar en un breve tiempo, gracias a los juegos, comparado al resto de métodos.

Los neurocientíficos cognitivos andan buscando juegos para probar estas hipótesis acerca de los avances cognitivos y los mecanismos concretos sobre los que se basan (Bavelier y Davidson, 2013). Además de estas habilidades espaciales, han especulado que los videojuegos son un medio excelente para el desarrollo de habilidades de resolución de problemas (Prensky, 2012).

Por ello en el laboratorio Heron de la Universidad de Montréal se me pidió realizar este juego. Buscan realizar pruebas cognitivas gracias a Hospital Amnesia.

Otro estudio (Adachi y Willoughby, 2013) mostró a través de un self-report, que los juegos estratégicos ofrecieron más mejoras en las habilidades de resolver problemas. Y a su vez, esto tuvo una consecuencia en su expediente académico, obteniendo mejores notas.

Un dato curioso, en 2008, investigadores de la Universidad de Washington crearon un juego online llamado Foldit (Cooper et al., 2010), que permitía a cualquier persona modelar la composición genética de las proteínas. Gracias a una competición de 3 semanas, se identificó una solución rápida de la estructura cristalina de un virus de mono relacionado con el SIDA. Se obtuvo gracias a la resolución de problemas de cooperación de los jugadores.

En otra investigación (Mejía, 2012) analizaron niños en un juego de plataformas, Toy Story, y se dedujo que los videojuegos podrían ser considerados como situaciones de resolución de problemas.

Estos investigadores crearon bajo el nombre de SPIDD (sistema psicométrico-informático para el diagnóstico del desarrollo) un videojuego para evaluar tres funciones: ejecutiva, memoria y atención. Dedujeron que los videojuegos comerciales, pueden ofrecer mucha información acerca del desarrollo cognitivo, que difícilmente podría obtenerse mediante el uso de herramientas y técnicas convencionales.

Varios estudios, (Basak, Boot, Voss, & Kramer, 2008) y (Anguera et al., 2013) descubrieron que no sólo mejora los procesos cognitivos en ancianos, sino que también ayuda a que no se degeneren. Siendo muy positivo el uso de los videojuegos en ancianos.

Las pruebas mostraban (Basak et al., 2008) como los pacientes ancianos, una media de edad de 70 años, mejoraban sus resultados en cada sesión realizada. El otro estudio (Anguera et al., 2013) utilizaba un videojuego modificado *NeuroRacer* buscando mejorar la multitarea, y al cabo de varios meses, estos pacientes de avanzada edad notaron una gran mejoría respecto a los jugadores más jóvenes.

Y por último investigadores italianos (Franceschini et al., 2013) demostraron que jugar 12 horas semanales a videojuegos de acción (sin contener ejercicios fonéticos u ortográficos) podía aumentar significativamente las competencias lectoras de los niños con dislexia.

Como se ha mencionado antes (Green & Bavelier, 2006), los juegos de acción puede aumentar la atención, pero estos resultados mostraron que esta atención también se traduce en una clara mejora para la dislexia, otorgando a este grupo mayor atención al leer, y así corregir este problema.

Estructura del proyecto

3.1 Objetivos generales

Este proyecto se ha realizado para el laboratorio Heron de la universidad de Montréal, Canadá. El objetivo es poder realizar estudios a pacientes sobre su cognición, sumergidos en videojuegos, y así ver su influencia de esta inmersión en la cognición de estos pacientes.

Para ello, se ha creado este video juego, Hospital Amnesia, con varios mini juegos dentro de la historia principal, para así realizar pruebas en varias áreas de la cognición.

Cada mini juego se ha creado buscando un área en específico. El videojuego crea un log file, que podrá consultarse sin problemas, para complementar el estudio que se realice con él. En él aparecen las acciones realizadas por el jugador, de forma que se puede saber dónde se encuentra el jugador y en qué situación se encuentra.

Uno de los objetivos, es que se pueda utilizar con Tobii, una herramienta para el seguimiento de los ojos (*eye tracking*) y poder analizar el razonamiento seguido por el paciente. Por ello, la interfaz de los mini juegos intenta estar repartida por toda la pantalla, puesto que este seguimiento puede ser algo sensible.

También será utilizado en conjunto al electroencefalógrafo (EEG) o el ritmo. Un EEG mide la electricidad de las células del cerebro que utilizan para comunicarse entre ellas.

Esta es la razón por la que el juego no puede durar más de una hora. El casco EEG, necesito un gel para conectar las ventosas al cuero cabelludo, que a la hora se seca e impide continuar con el seguimiento, dando resultados incompletos o nulos.

El videojuego tiene un entorno 3D en un hospital, que es la parte visualmente más atractiva, en la que explicamos al jugador cómo funcionan los juegos y qué objetivos tiene que alcanzar, se le guía a través de todos los juegos y la historia.

Los mini juegos son en 2D, algunos son juegos como tal, y otros son simplemente test's que te puedes encontrar en cualquier web.

El proyecto tendrá dos partes bien diferenciadas:

Diseño:

El diseño es todo lo que el usuario verá a lo largo del juego y se realizará al principio, antes de crear los scripts necesarios. Necesitamos un entorno antes de saber que programar.

Programación:

La programación se realizará en el desarrollo y se irá adjuntando a objetos o al personaje mismo. Se encargará del funcionamiento del juego, sin ella, no se podría interactuar en absoluto con ningún objeto, ni siquiera mover al personaje.

3.2 Metodologías

Siendo el método que más se ajusta a este proyecto, se usará el Proceso Unificado de Desarrollo de Software que se basa en la implementación de casos de uso y UML.

El Proceso Unificado es un marco de desarrollo iterativo e incremental, en el cual tendremos una primera fase de análisis, en la que analizaremos los requerimientos necesarios por cada parte de juego (es decir el principal y los mini juegos).

La segunda parte constará del diseño, dónde diseñaremos la parte visual de estos juegos, así como su funcionamiento y scripts que son necesarios.

La tercera, será la parte de desarrollo, en la que se programará todo el código de los objetos y del personaje principal.

Y por último siempre se realizarán pruebas hacia el final de su desarrollo, el desarrollo no habrá acabado hasta que las pruebas se hayan superado.

Cada iteración constará de estas 4 fases, que serán iterativas, puesto que se irán añadiendo nuevas funcionalidades o nuevos juegos, al principal juego.

3.3 Tecnologías

Los programas utilizados serán los siguientes:

Diseño:

- **Unity 3d:**
Herramienta para el diseño de videojuegos con un potencial increíble y gratis para desarrolladores independientes. A parte de ser bastante fácil de utilizar, contiene todo lo necesario para crear un videojuego funcional.
- **Blender:**
Software para la creación de objetos 3D, similar al famoso 3Ds Max, a partir de ciertos modelos o desde 0.
- **Mixamo Fuse:**
Software para personalizar los personajes, animaciones y vestimenta.

Programación:

- **MonoDevelop** (Unity 3d): Integrado en Unity 3d, es el IDE para escribir el código, compilarlo e integrarlo posteriormente al propio Unity. No ofrece tantas posibilidades como otros IDE (véase Visual Studio) pero al venir incorporado con Unity es la forma más cómoda de escribir nuestro scripts, aparte de que los lenguajes utilizados no son realmente JavaScript y c#, sino un framework. Sirve tanto para UnityScript como C#, pero sólo se ha utilizado C# en este juego.
- **Notepad ++:** El famoso bloc de notas, ha servido para escribir los xml que se leen en los minijuegos de lógica y diagnóstico.

Otros:

Página web externa: www.lumosity.com

3.4 Planificación

Semanalmente se realizaban reuniones con Asma Ben Kheder y Claude Frasson, director del laboratorio Heron, para modificar o añadir estos juegos.

La primera fase fue el análisis, qué necesitaban para realizar su estudio. Se buscó una temática en la que se pudiera probar y realizar pruebas médicas, y qué mejor, que un hospital. Con la idea de qué el jugador padece de amnesia, se le realizarán pruebas relacionadas con los procesos cognitivos. Primero se creó un entorno 3d, con sus consecuentes fases de diseño, desarrollo y pruebas, en esta se probaría la resolución de problemas así como sumergir al jugador en un entorno interactivo.

Después con la misma idea y con la misma planificación, se idearon varios minijuegos buscando analizar una o varias habilidades cognitivas. Los profesores querían un juego para médicos, buscando utilizarlo con estudiantes de medicina o incluso médicos en los cuales su razonamiento cognitivo se ha deteriorado con el paso de los años.

Asma, buscó páginas web en las cuales hubiera test's para ampliar su estudio, y encontró www.lumosity.com que se integró con el juego, recopilando los resultados.

Se quiso también crear un juego lógico, que consiste en varias preguntas típicas que se encuentran en cualquier web, pero estos ejercicios se decidieron integrar directamente en el juego.

Y por último el juego de memoria se dejó para el final puesto que al comprarse, sólo se necesitaba de unas pocas modificaciones para incorporarlo en el juego.

Por último, se siguen realizando correcciones de errores sin importancia, todos relacionados con el diseño. Pequeños fallos de objetos mal situados o incluir alguna animación nueva.

Y por tanto las pruebas siguen hasta el final de todas estas correcciones y hasta el día de la fecha final del proyecto.

La supervisión final vendrá de la parte de Baltasar Fernández Majón, profesor de la Universidad Complutense de Madrid.

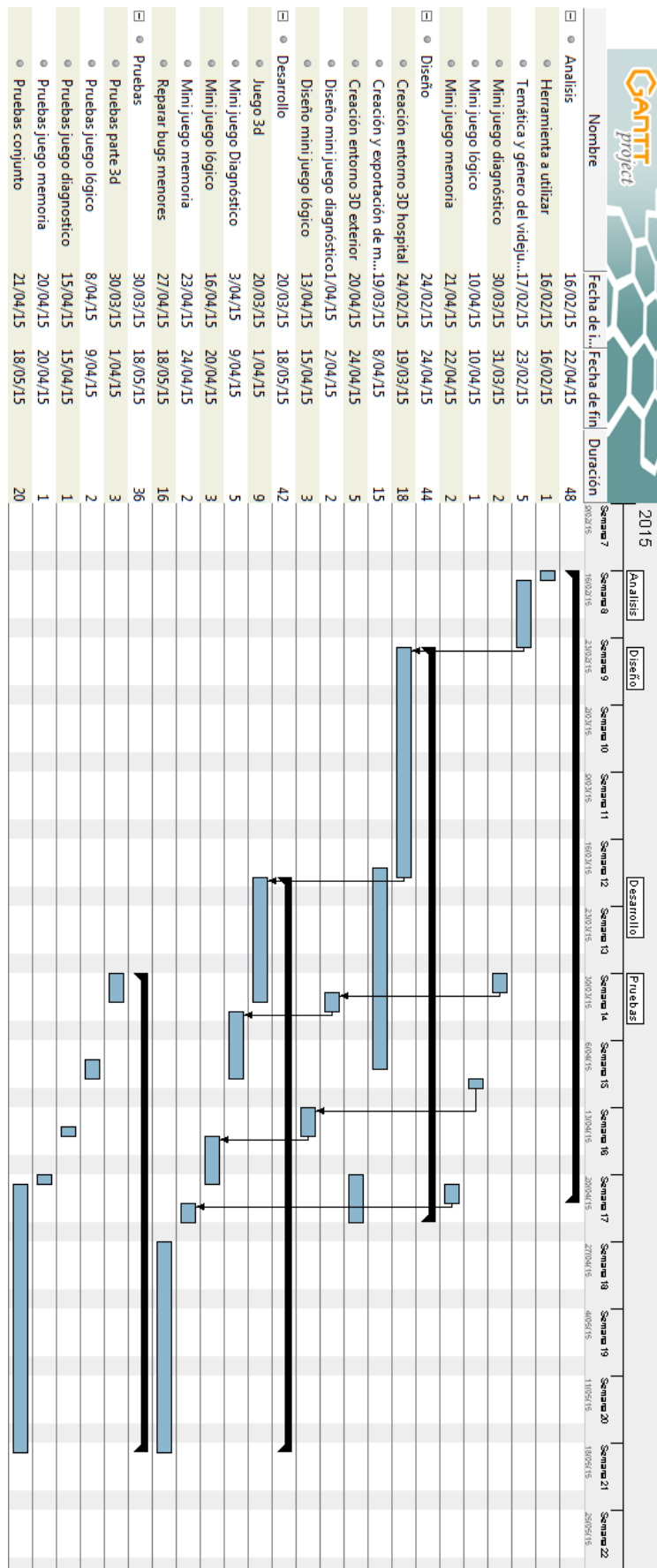


Ilustración 4. Diagrama de Gantt.

Arquitectura

El programa utilizado ha sido Unity 3d, uno de los mejores programas para desarrollar juegos independientes. Las funciones básicas, que no son pocas, son gratis, y el potencial es increíble. Puedes desarrollar mundos 3d en poco tiempo y con el apoyo de su tienda virtual aún más rápido, algunos paquetes que mencionaré más adelante se compraron para agilizar el proyecto.

El lenguaje utilizado ha sido C#, algo más potente y con más posibilidades que UnityScript (framework de JavaScript para Unity).

La versión de Unity utilizada ha sido la versión 4.6.2f1. La versión del juego final será funcional para Windows 32 bits y 64 bits.

4.1 Estructura del videojuego

Unity 3D contiene 5 vistas importantes, que vemos en la imagen.

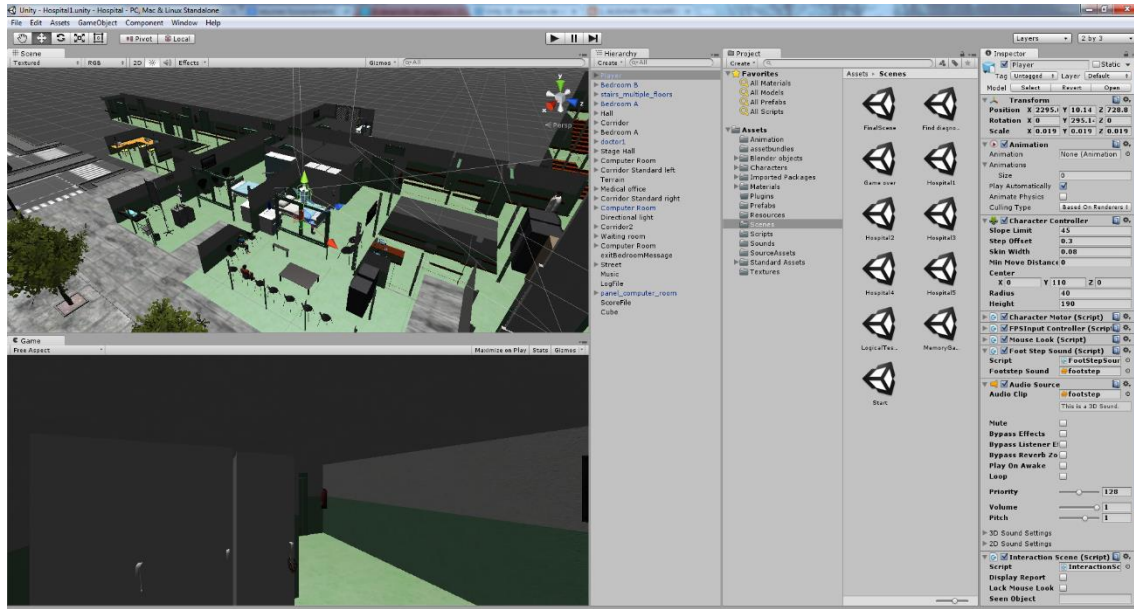


Ilustración 5. Vistas de Unity 3d.

La **escena**, dónde diseñamos la parte visual, es decir todo el escenario del nivel en el que nos encontramos. Arrastramos los objetos desde la zona de proyecto, para añadirlos.

El **juego**, lo que ve nuestra cámara, en este caso el personaje. Podemos ir diseñando en la escena mientras vemos el resultado final.

La **jerarquía**, dónde aparecen todos los objetos que contiene la escena en la que nos encontremos. Cada objeto puede ser un conjunto de más objetos. Podemos crear incluso un GameObject vacío y añadirle todos los objetos que queramos, formando un GameObject más complejo. En el proyecto he intentado organizar una jerarquía de una forma bastante lógica. En primer lugar he dividido la escena en habitaciones, de forma que tengamos Bedroom A o B (dependiendo su orientación), pasillos, entradas, salas de espera y otras salas.

Cada estancia a su vez está compuesta de más objetos, algunos compuestos, otros simples. Todas llevan el mismo suelo y techo, que es un objeto en sí. Las puertas suelen ser objetos compuestos también, diferenciando el marco, el pomo o la puerta en sí.

El **proyecto** contiene los assets. Los assets son las carpetas donde guardamos todo tipo de elementos. En este proyecto, se han creado algunas carpetas muy claras para tenerlo bien organizado.

- **Blender Objects:**
Se guardan todos los objetos generados por Blender.
- **Characters:**
Todos los personajes que aparecen y se han creado por Mixamo y Blender, están almacenados aquí.
- **Imported Packages:**
Todos los paquetes comprados se importan a esta carpeta, muy importante, puesto que a la hora de subir el código a GitHub, no se deberá subir esta carpeta, por asuntos legales. No podemos distribuir libremente estos assets.
- **Materials:**
Se guardan todas las texturas que usaremos en el juego. Las texturas son los colores o capa que se le añade a un objeto.
- **Prefabs:**
Se almacenan todos los prefabs. Los prefabs son un tipo especial de objeto.
- **Resources:**
Todos los ficheros que se tienen que leer durante la ejecución del juego están aquí
- **Scenes:**
Los niveles que creamos en el proyecto se guardan aquí.
- **Scripts:**
Contiene todos los scripts del juego que se utilizan.
- **Sounds:**
Sonidos que se usan en el juego, es decir música y grabaciones de audio.

Y por último el **inspector**, una vista que al seleccionar cualquier objeto, te muestra todos los componentes asociados a este. Podemos añadir directamente desde ahí nuevos componentes, o arrastrar hasta esta zona. Los componentes son funcionalidades, como scripts, texturas, posición, colliders, etc...

La herencia en Unity es importante conocerla, para saber qué límites tenemos y qué funcionalidades se pueden usar en cada objeto. En la Ilustración 5 (Extraída de <http://unityscripts.blogspot.com.es/2011/10/algunas-peculiaridades-de-unity.html>) se puede ver la herencia.

Aunque no todos hereden de Component, se les llama componentes. Es lógico que el material no herede de Component puesto que no necesitan una luz o un collider por ejemplo.

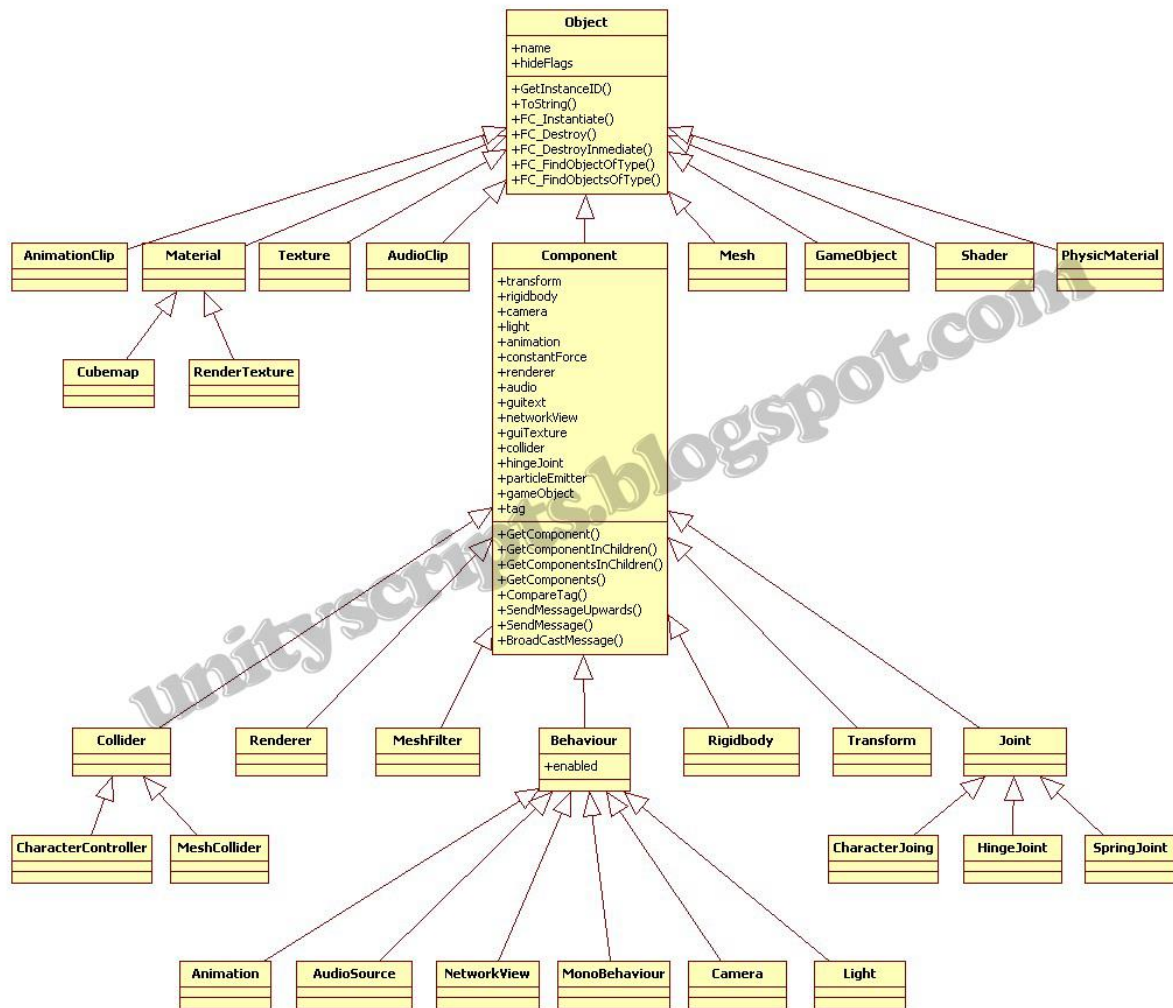


Ilustración 6. Herencia en Unity 3d.

En los siguientes capítulos, veremos los componentes y elementos más relevantes. Empezando por los objetos en sí mismo.

4.2 Modelado 3D

4.2.1 Paquetes comprados a través de la asset store de Unity 3d

Unity pone a disposición una tienda de paquetes (o llamados assets), como muchos software o sistemas operativos, cuya dirección es <https://www.assetstore.unity3d.com>.

El laboratorio compró varios paquetes, se intentó sólo comprar los necesarios para el desarrollo del juego y utilizar los gratuitos si estaban disponibles, pero nunca hubo problemas para obtener el paquete que hiciera falta. Estos paquetes tienen una licencia especial de Unity, de forma que no se puede reproducir, duplicar, copiar, vender, comercializar o revender, indicado en el punto 3.5 (http://unity3d.com/legal/as_terms). Por tanto se indicarán los paquetes utilizados, explicando su uso, y nombrándolos con el nombre que aparecen en la tienda de Unity, de manera que si alguien quiere usar este proyecto, deberá adquirir estos paquetes.

Estos paquetes serían los siguientes,

Primero, los de pago:

Generic Modular Modern Construction Kit:

Uno de los paquetes más importantes, se ha utilizado para construir todo el complejo hospitalario. Contiene muros, ventanas, puertas, escaleras para usar en la construcción, así como materiales.

Hospital Props:

Contiene varios objetos médicos que se han utilizado en la oficina del médico, alguno más en el resto del hospital.

Medical Clothing Pack:

Toda la ropa que se ha usado para vestir a médicos y enfermeros, se usó en Mixamo.

Hospital Room Set:

Contiene la cama, una mesilla y el armario de las habitaciones del hospital.

Desktop Computer:

Es el ordenador utilizado en todo el hospital, puesto que en algunas escenas es prácticamente lo único que hay, tenía que estar bien diseñado.

Memory Game Starter Kit:

Juego de parejas de cartas que se ha modificado posteriormente para cumplir los objetivos buscados.

Y a continuación, los gratuitos:

Male character Pack:

Personaje utilizado para movernos en el juego, aunque no se le ve, desarrollado por Maximo.

NYC BLOCK #6:

Entorno utilizado para modelar todo el entorno exterior del hospital, creando una pequeña plaza con edificios y el hospital.

Wall TVset:

Televisión plana que está situada en los cuartos del hospital.

Bathroom props:

Objetos utilizados en los baños de las habitaciones de los enfermos.

Snack Machines:

Máquinas expendedoras de bebidas y comida, para salas de esperas o halls.

Vending machine:

Máquina expendedora de bebidas, mismo propósito que con las snack Machines.

Free Furniture Set:

Mesa y sillas utilizadas en la sala de ordenadores.

Free Furniture Pack 1:

Mesa modificada para usar en los halls o salas de espera.

HackSaw:

Sierra que se modificó en Blender posteriormente.

4.2.2 Objetos añadidos o modificados por Blender

Los objetos podían venir de cualquier fuente, en nuestro caso de 3 principales, Unity, Mixamo y externo (web), como objetos creados en 3ds Max y sin coste a disposición de cualquier internauta.

Personas:

Todas las personas han pasado por el programa Mixamo, para vestirles. El paquete que contiene ropa para los modelos se añade a este programa, y con él podemos vestir a nuestros personajes. Nos ofrece dos personas a las cuales se les puede personalizar, modificando partes de su cuerpo, y añadiéndoles ropa. A parte del personal médico, también se aprovechó la ropa que traía de base el programa para crear algún paciente con ropa de calle.

Hacksaw:

Adquirido en Unity, se dividió en dos objetos independientes con Blender. Para repartirlos por la habitación.

BigCabinet:

Es un objeto descargado de un paquete anterior, al cual se ha dividido también dos partes, y juntado en Unity para aplicarle la animación que lo abre.

Chainsaw:

Cadena modelada con Blender, se fueron juntando aros hasta formar una cadena casi circular para colocar en el armario.

Números:

Todos los números que aparecen en el juego son generados a partir de Blender, teniendo cada uno un material que se le puede asociar, que al final suele ser simplemente un color.

Carteles:

Algunos carteles que aparecen (como el de “exit”) se han creado también gracias a Blender.

Persianas y botellas:

Importadas desde la web <http://archive3d.net/> que pone a nuestra disposición miles de objetos en formato 3d gratuitamente. Todos los objetos en este tipo de formato (.3ds) se pueden incluir en Unity sin problemas.

4.3 Scripts

Todos los scripts se han creado en C#, se ha tenido que traducir algunos que venían de serie en JavaScript a C#. Casi todos son independientes y van asociados a un objeto en cuestión. Estos objetos llamados `GameObject` pueden tener tantos scripts como uno desee. La comunicación entre ellos es muy limitada. Esta comunicación sobre todo se realiza entre objetos (pueden ser triggers por ejemplo).

Es muy importante nombrar ciertas funciones que contiene cualquier script, que vienen por defecto, pueden estar vacías o no. El juego las ejecuta en determinados momentos. Las más importantes y algunas utilizadas son (“Monobehaviour”):

Awake():

Se llama cuando se carga la instancia.

Start():

Es el primer método en llamarse en el primer frame, para configurar el objeto, antes de llamar a cualquier otra función de update.

Update():

Si el `MonoBehaviour` está activo, se llama en cada frame.

OnGUI():

Se utiliza para la gestión de eventos GUI.

OnTriggerEnter(Collider OtroColisionador):

Se llama cuando este colisionador / rigidbody justo empieza a tocar otro colisionador/rigidbody.

OnTriggerExit(Collider OtroColisionador):

Se llama cuando este colisionador / rigidbody justo termina de tocar otro colisionador/rigidbody.

OnTriggerStay(Collider OtroColisionador):

Se llama cuando este colisionador / rigidbody mientras está tocando otro colisionador/rigidbody.

OnCollisionEnter(Collider otro):

Se llama cuando el colisionador otro entra en el *trigger*.

OnCollisionExit():

Se llama cuando el colisionador otro sale del *trigger*.

OnCollisionStay():

Se llama mientras el colisionador está en el *trigger*.

GetComponent():

Devuelve el componente del tipo Tipo si el gameObject tiene uno adjunto, null si no lo tiene.

DontDestroyOnLoad():

El objeto con este script asociado no se destruirá con la carga de una nueva escena.

Los scripts utilizados en el juego son los siguientes:

Character Motor:

Es el script asociado que mueve al jugador. Unity viene de serie con este script pero en UnityScript. Se ha traducido a C# para que pueda comunicarse con el resto.

Credits:

Se adjunta al collider para una vez que los créditos (es un objeto a efectos prácticos) entre en éste, se cargue la escena de inicio.

DoctorMessage:

Asociado al primer doctor, muestra un mensaje y activa el audio del personaje cada vez que entra en el trigger.

DoctorMessage2:

Idéntico al anterior.

exitBedroomMessage:

Una vez sale del dormitorio, se muestra un mensaje.

FinalMessage:

Mensaje que se muestra al final del juego en la calle.

FootStepSound:

Cada vez que el jugador avance o retroceda, el sonido de stepSound, sonido de pasos, se activa, simulando el sonido del jugador andando.

FPSInput Controller:

Complementario al carácter motor, reconoce el teclado. También se tuvo que traducir.

GameController:

Script principal del mini juego de cartas. Se ha modificado para integrarlo al juego, de forma que las parejas se incrementen en cada vuelta en una, y al jugar 3 rondas se cargue directamente la escena siguiente.

GameManager:

Se encarga de gestionar el juego de los diagnosticos, se le adjunta a la cámara y se ejecuta como GUI (interfaz de Unity). Carga de varios ficheros xml los casos, y rellena los botones o casillas.

GameManagerLogicalTest:

Gestiona el mini juego de preguntas lógicas, carga también los ficheros de archivos para cada pregunta.

GameOver:

Script que te indica que has perdido y tienes que quedarte en el hospital.

GameOverCollider:

Se adjunta al collider para una vez que las letras de Game Over (es un objeto a efectos prácticos) entre en éste, se cargue la escena de inicio.

InteractionScene:

Script principal del juego, va asociado al jugador en todo momento en las escenas 1 y 2. Permite que el jugador actúe con objetos, mostrar varios mensajes durante el desarrollo, así como comprobar y mostrar las combinaciones. El script MouseLook consultará este para bloquear el jugador en esos casos. Cada objeto lo tratará de formas distintas, algunos simplemente los recogerá, otros mostrará una combinación, o los ordenadores, interactuará, ejecutando un juego o lanzando una web.

InteractionScene3:

Solo se usa en la escena 3, muy básico, se deja de usar el anterior, que era mucho más pesado y realizaba en cada frame muchas combinaciones, y usamos este para aligerar la carga. Simplemente interactúa con el último ordenador.

InteractionScene5:

Idéntico que el anterior, pero en vez de actuar con el ordenador, lo hace con el paciente tumbado en la camilla.

loadScene2:

Se adjunta a un trigger en el cual al entrar cargue la escena 2.

LoadScene5:

Ídem que anterior, cargando la escena 5.

LogFile:

Script que se asocia a un objeto indestructible durante todo el juego, para no perderlo en ningún momento, logfile que irá registrando las acciones del jugador con hora y fecha exacta.

MouseLook:

Script que viene de serie, pero modificado para comunicarse con el script InteractionScene, de forma que si la variable lockMouse es cierta, bloquee todo movimiento.

Music:

Asociado al jugador o la cámara, reproduce en bucle música de fondo.

NurseMessage:

Mensaje que se muestra y se escucha, cuando te acercas a la enfermera.

NurseMessage2:

Ídem que anterior, cambiando el mensaje.

OpenDoor:

Asociado a la puerta que se quiere abrir pulsando “E”.

Patient1Message:

Mensaje que se muestra cuando te acercas al paciente.

Patient2Message:

Mensaje que se muestra cuando te acercas al otro paciente.

PlatformInputController:

Permite que el jugador avance en la dirección hacia la que mira, también venía de serie y se tradujo.

PlayerInput:

Necesario en escenas que no esté el jugador “físicamente” para poder mover y usar el puntero, en escenas como mini juegos o la pantalla de inicio.

ScoreFile:

Otro logfile que solo registrará las puntuaciones en los juegos que se puntúe.

StartGame:

Se utiliza en la escena inicial que se muestra al iniciar el juego. Simplemente carga la primera escena o te saca del juego.

WelcomeScene2:

Mensaje que se muestra al entrar en el cuarto de ordenadores, indicando que tienes que realizar.

WelcomeScene4:

Mensaje que se muestra al entrar en la oficina del doctos, y también te da instrucciones.

WelcomeTrigger:

Primer mensaje que se muestra en el juego, explicándote la situación en la que te encuentras, y que objetivos tienes que conseguir. Y una breve ayuda para moverte por el entorno.

4.4 Escenas

Las escenas son niveles de Unity. Pueden estar conectadas entre sí o no. Aunque la idea de estas escenas es conectarlas para crear una serie de niveles y así el juego. Cada escena suele tener objetos independientes, pero existe la posibilidad de no destruir los objetos al cargar un nuevo nivel y conservarlo. Es el caso de la música o los logfile, se crean al inicio del juego, y no se destruyen hasta el final gracias a la función DontDestroyOnLoad().

Este juego no es completamente lineal, existe la posibilidad de saltarte una serie de escenas (y por tanto mini juegos) si tus resultados en los ejercicios de la web externa son mejores que 70 sobre 100.

Cada mini juego se trata como una escena, siendo completamente independiente, y no sobrecargando una escena.

En la ilustración 6 se puede ver el funcionamiento de las escenas.

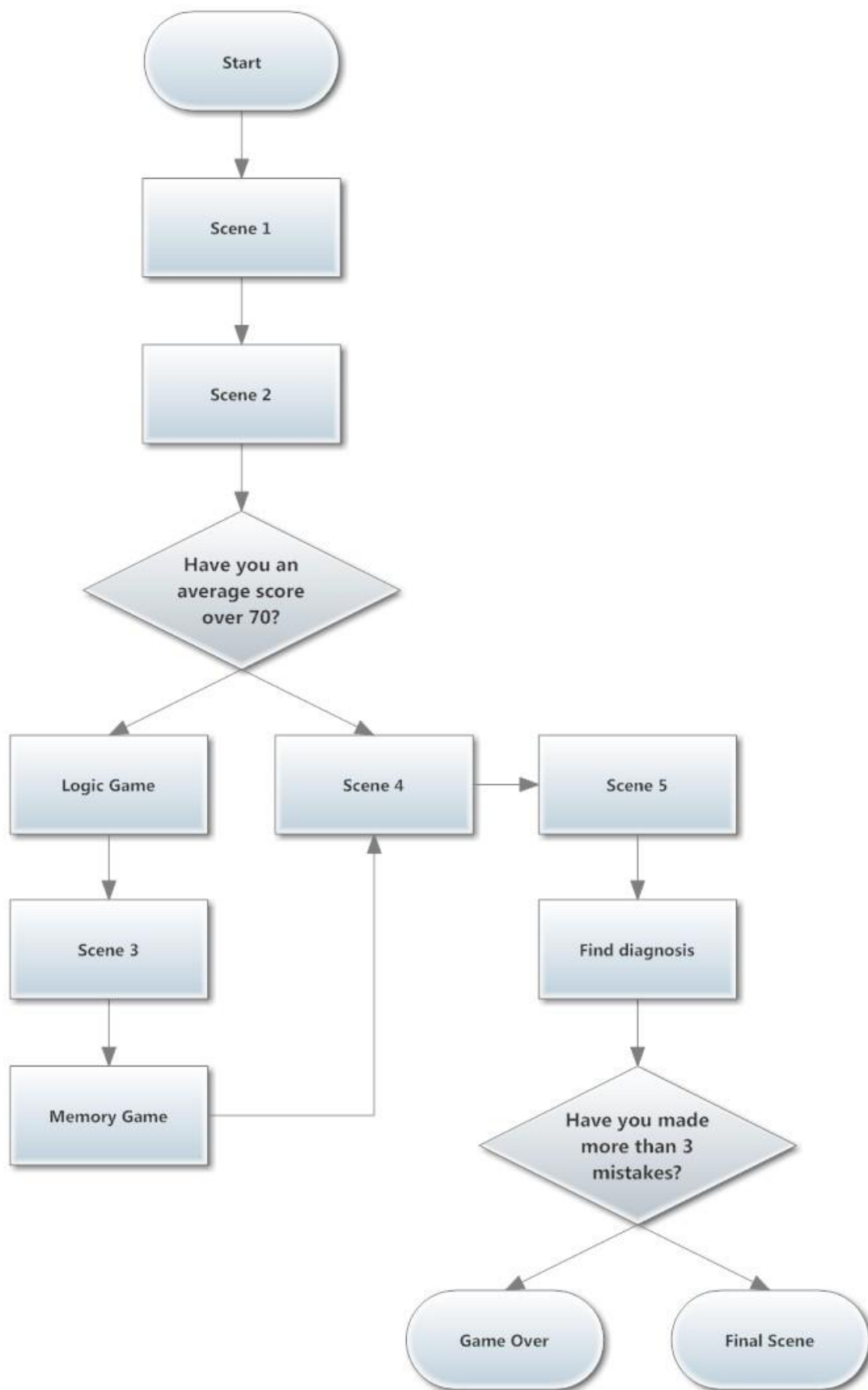


Ilustración 7. Diagrama de flujo de las escenas.

4.5 Animaciones

Existen varias formas de animar los objetos, desde un programa exterior como Mixamo, o simplemente con Unity. La forma más sencilla de animar un objeto es mediante la herramienta de Unity.

Para ello se elige el objeto a animar, se pulsa “ctrl + 6”. Se abrirá una ventana para crear la animación. Se elige la posición que queremos que ocupe el objeto en cada frame, y Unity se encarga de realizar el movimiento directo entre ambos frame.

Es decir, si colocamos la puerta abierta en un frame y la puerta cerrada en otro, Unity creará el movimiento de la puerta para cerrarla o abrirla.

Esta animación se guarda y se adjunta al objeto. Se puede ejecutar nada más iniciarse, y entrar o no en bucle. Pero a nosotros no nos interesa eso, queremos que la animación se active en determinados momentos, ya sea pulsando una tecla o acertando la combinación.

En la Ilustración 7 tenemos un ejemplo de cómo realizar la animación de una puerta. Se coloca la puerta en una determinada posición y se inserta el frame en la línea de tiempo de abajo. Hacemos exactamente lo mismo con la otra posición, insertando el frame en otro momento de la línea. En este caso se realizaría una simple rotación de la puerta.

Dependiendo en qué segundos indiquemos las posiciones, la animación será más lenta o más rápida, cerrándose (o abriéndose) la puerta a una u otra velocidad.

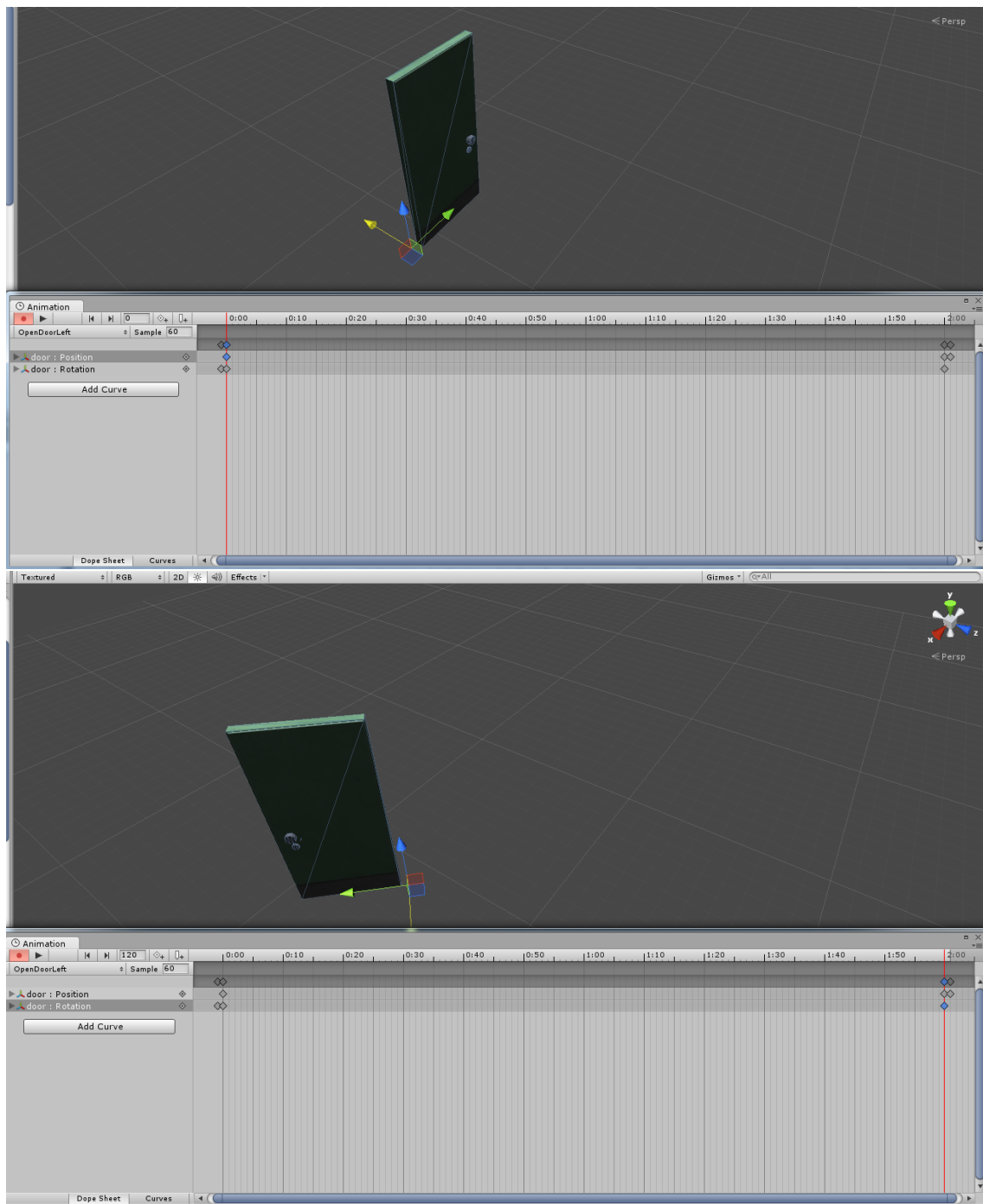


Ilustración 8. Animación en Unity 3d.

4.6 Audio

Se puede incorporar cualquier tipo de audio en el juego, desde música a efectos de sonido. El audio tiene que ir asociado a algún objeto, como cualquier componente.

Se necesita un AudioListener en la escena, que suele ir adjunto a la cámara o el jugador, es el componente que “escucha” el audio, sin él el audio no se escucha al reproducirse.

Sólo debe haber uno, o puede haber conflictos en la escena.

En cambio AudioSource, puede haber ilimitados. En nuestro caso los usamos con 3 propósitos:

Música de ambiente: La música de ambiente se adjunta a un objeto indestructible llamado Music, de forma que se escuche durante todo el juego. Se ha creado una opción en el menú de Pausa, para poder cambiar el volumen de esta música. La música de ambiente utilizado se ha descargado de la librería de música de youtube, <https://www.youtube.com/audiolibrary/music> . Está a disposición de cualquier internauta, normalmente para usar en los videos de youtube, pero se ha usado para el juego también. La música utilizada se llama, New Land.

Efectos de sonido: Se ha añadido el sonido de los pasos al jugador cuando se mueve.

Conversaciones: Se han grabado los diálogos de los personajes que hablan. El doctor lo dobla Sergio Primo y la enfermera Asma Ben Kheder.

4.7 GUI

Quiero dedicarle un capítulo a la interfaz, puesto que todo el funcionamiento de los mini juegos y mensajes que aparecen a lo largo del juego es gracias al GUI.

La interfaz gráfica de usuario, conocida más como GUI, actúa como interfaz de usuario, utilizando un conjunto de objetos gráficos tales como botones, paneles o incluso controles deslizantes. Estos pueden modificarse de muchas maneras, pudiendo cambiar fuentes, añadir imágenes, bordes, etc... A algunos elementos, como los botones, se les puede además añadir una funcionalidad.

He utilizado las variables `Screen.height` y `Screen.weight` para ajustar la interfaz según la pantalla, en vez de variables, puesto que cada pantalla tiene una resolución distinta y hay que ajustar el GUI a estas. Si usamos constantes, se utilizan los pixeles, y no todas las pantallas tienen las mismas dimensiones (desde 800x600 a 1960x1080).

La GUI se puede configurar de muchas maneras. Está creada por defecto y es única. Por tanto cada vez que realicemos alguna modificación en la GUI, se conservará en el resto de la escena, a no ser que la volvamos a modificar.

Es decir, si cambiamos el tamaño de la fuente directamente en el GUI, el tamaño será este para los textos que sigan.

Hay una excepción para los que tengan un style definido, en adelante explicaré qué son.

Todos los mensajes que aparecen durante el juego se realizan mediante la GUI. Se utiliza la función `GUI.label(posición, mensaje)`.

Por eso, se usan variables booleanas para mostrar los mensajes, de forma que activamos la variable oportuna para mostrar el texto que queramos, y la desactivamos cuando queramos dejar de mostrarlo. Esto se debe a que el GUI tiene el mismo funcionamiento que el `Update()`, del que hablamos en el capítulo de Scripts, que se llama continuamente.

Para crear distintos tipos de formatos para los paneles o botones, he utilizado Styles. Se crea un style de tipo `GUIStyle`, que se puede modificar incluso más que el GUI.

En el juego a cada style le indicaba el tamaño de letra, los colores y su alineamiento. Además dependiendo su uso se le podía cambiar más el formato. Se pueden añadir cuando se crea cualquier elemento como entrada en la llamada.

Un ejemplo de Style podría ser el siguiente:

```
buttonReadyStyle = new GUIStyle ();
buttonReadyStyle.fontSize = 30;
buttonReadyStyle.alignment = TextAnchor.MiddleCenter;
buttonReadyStyle.normal.background = buttonReadyBackground;
```

Este estilo está asociado al botón verde que aparece en la Ilustración 8. `buttonReadyBackground` es una imagen, en este caso, es el fondo verde con bordes negros.

En la imagen se pueden ver otros estilos, como el de los botones azules o la información que aparece en las libretas de médico.

En este mini juego la interfaz serían las 3 libretas, los 3 botones azules y el verde. El fondo es una escena 3d que se realizó con ayuda de los elementos ya disponibles del entorno 3d, y cuadrando la cámara.

Los botones suelen llevar una funcionalidad, esta se define con la variable de condición, podemos ver cómo sería el código del botón verde:

```
if (GUI.Button (new Rect (2 * Screen.width / 3, Screen.height /  
8, Screen.width / 4, Screen.height / 6),  
"Are you ready for the diagnosis?\n Click here.",  
buttonReadyStyle)) {  
  
    //FUNCIONALIDAD  
  
}
```

En este juego, cuando se elige una opción, cambia el color de la fuente, y esto se realiza, en el apartado de funcionalidad cambiando el color de la fuente.

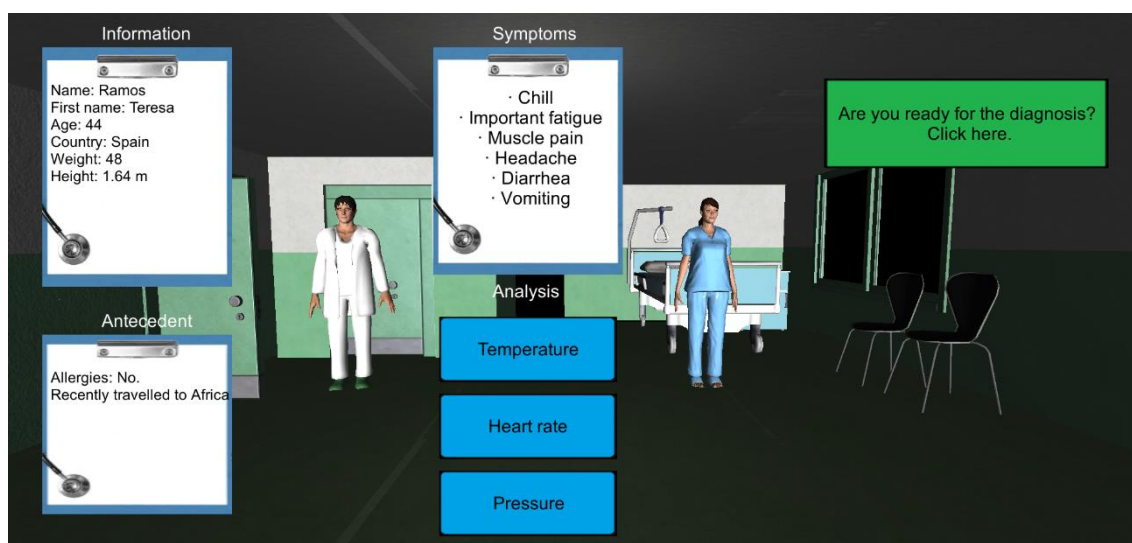


Ilustración 9. Mini juego diagnóstico.

Pruebas

Las Pruebas se han realizado a lo largo de todo el proceso desde el momento que se ha escrito y asociado un script. Se realizan cada vez que se corrige o cambia algo en la escena.

Unity nos facilita estas pruebas gracias al compilado automático del código, y al botón Play, que ejecuta inmediatamente la escena en la que estás trabajando.

Las más representativas fueron después del desarrollo del entorno 3d y de cada mini juego.

5.1 Entorno 3d

Fueron las pruebas que más tiempo tomaron, puesto que fue la parte más importante a desarrollar. Una vez que el entorno se estaba acabando, empezaron las pruebas con el personaje.

Se desplazaba el paciente por el entorno 3d, así se podía ver que objetos estaban o no bien diseñados. Puesto que lo que se veía, era lo que el jugador final vería.

Los muros se tenían que colocar bien, es difícil cuadrar exactamente los objetos en el desarrollo, al ser más pequeños, por ello se necesitaba la visión en primera persona continuamente.

Una vez los objetos estaban bien situados y su aspecto era el deseado, se empezaron a añadir componentes, los más importantes los scripts y los *collider*.

Cada vez que se creaba o modificaba un script, se realizaban pruebas sobre éste. Nunca se modificaba sin después realizar las pruebas, para no ir acumulando código y en caso

de error no tener que depurar el script entero, que el principal del personaje cada vez se complicaba más.

Los *collider* fueron un problema, puesto que el usuario podía parcialmente atravesar algunos muros y ver el esqueleto, o simplemente a través, y no necesitar realizar algunas tareas anteriormente, sin hablar de lo poco estético que resulta.

Se probaron a base de recorrer el escenario una y otra vez, intentando atravesar todos los muros y puertas. Cada vez que se encontraba uno que se pudiera atravesar, se anotaba e inmediatamente después del recorrido del personaje, se cambiaba. Y se volvía al mismo muro comprobando que estaba arreglado.

5.2 Mini juego Lógico

Al depender del GUI, era muy importante probarlo cada vez que se realizaba un cambio, porque a diferencia del entorno 3d, que más o menos se ve cómo se construye, el GUI no.

Al ser una interfaz, no se podía a medida que se programaba, visualizar el resultado. Por tanto había que realizar continuamente pruebas con cada cambio, para que los botones o los paneles no se solapen. O simplemente que las letras encajarán y fuera la fuente adecuada.

Las posiciones de cada elemento se realizaba “a ojo” por tanto había que repasar continuamente que los elementos se situarán en la posición que se diseñó anteriormente.

Otras pruebas que se realizaron, fue al cambiar los textos por varios ficheros, de forma que se cargarán un fichero por caso y así poder modificar el contenido de los ejercicios externamente sin tener que conocer el código. El problema principal fue tener cuidado con la ruta de los ficheros, puesto que al compilar las carpetas cambian, y hubo que tener en cuenta ello.

5.3 Mini juego de diagnósticos

Con un funcionamiento parecido al juego lógico, basándose en GUI, las pruebas eran muy importantes y constantes. La complejidad de este juego era mucho mayor que la del anterior. Por tanto, después del entorno 3d, fue dónde más pruebas se realizaron.

Además los cambios de diseño de este juego fueron numerosos, resultando una mayor carga de pruebas.

5.4 Pruebas generales

Estas pruebas no pararán hasta la entrega final del proyecto. La entrega al laboratorio Heron se realizó el 28 de abril, pero el desarrollo no se ha detendrá hasta la entrega para la Universidad Complutense de Madrid, sin contar la posibilidad de que el juego se siga ampliando posteriormente.

Se realizan partidas enteras comprobando todo lo posible e intentando cambiar la forma de hacerlo. Gracias a la ayuda de sujetos externos, que no han tenido ninguna implicación en el desarrollo de éste, se ha podido probar el juego desde un punto objetivo.

Manual de usuario

6.1 Controles

La siguiente tabla muestra los controles del jugador.

TECLA	FUNCIONAMIENTO
W	MOVIMIENTO HACIA DELANTE.
S	MOVIMIENTO HACIA ATRÁS.
A	MOVIMIENTO HACIA LA IZQUIERDA.
D	MOVIMIENTO HACIA LA DERECHA.
E	ABRIR PUERTAS.
P	PAUSA.
R	MOSTRAR REPORTE.
ESC	PAUSA.
BARRA ESPACIADORA	SALTAR.
CLICK IZQUIERDO	INTERACCIÓN CON OBJETOS.

Tabla 1. Controles del usuario.

En la imagen se puede ver un objeto, con el que se puede interactuar. Hay que tenerlo en el punto de mira, es decir en el centro de nuestra pantalla, para interactuar con él. En este caso lo cogeremos.

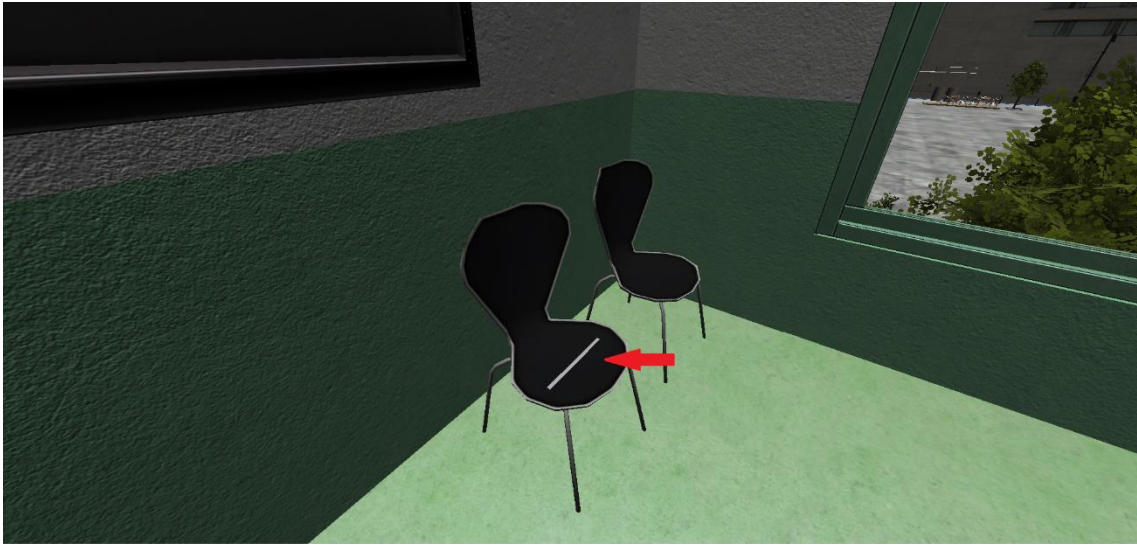


Ilustración 10. Interacción con objetos.

6.2 Menú de pausa

Un menú sencillo al pulsar la tecla de escape o P.



Ilustración 11. Menú de pausa.

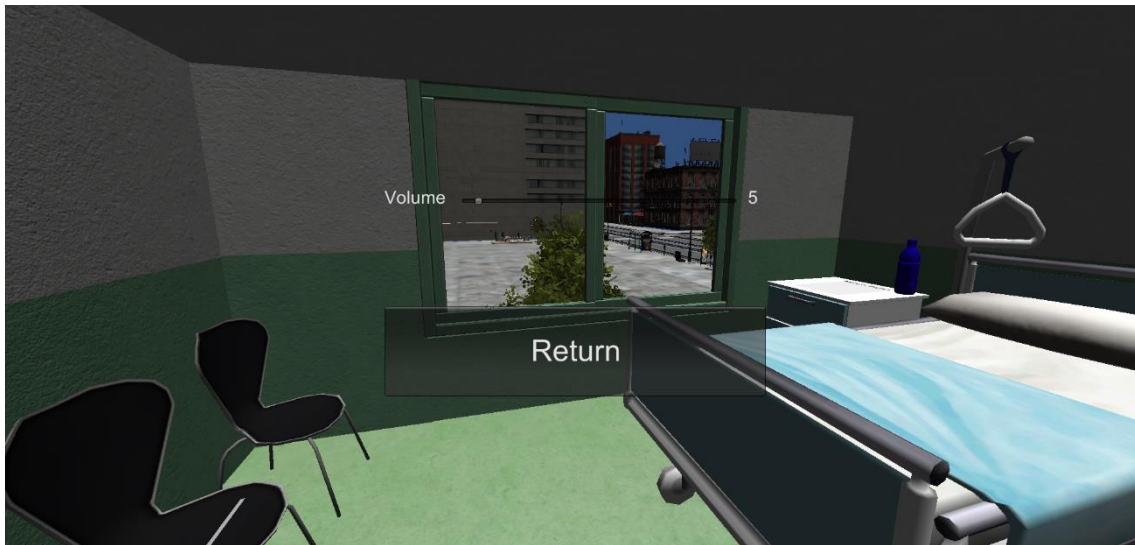


Ilustración 12. Menú para el control del volumen de la música.

6.3 Mini juegos

En estos juegos, el único control que se necesita, es el ratón. Simplemente se trata pulsar los botones con un click izquierdo.

Conclusión y trabajo futuro

7.1 Conclusiones y principales aportaciones

El juego desarrollado con un propósito claro para un estudio científico, intenta abarcar la mayoría de los tipos de cognición. Ha sido un proceso difícil al principio, al no conocer las herramientas, ni los esquemas de diseño de un videojuego. Muchos foros, el de Unity en especial (“Unity 3d forum”) (“Unity answers”) y una gran cantidad de videos, han ayudado a entender los conceptos y solucionar los problemas que surgían. Desde luego este motor es cada vez más utilizado, que se demuestra con el continuo crecimiento de su comunidad, muy importante para desarrolladores independientes.

El potencial de Unity y su tienda virtual de paquetes, ha agilizado mucho el proceso, pudiendo avanzar y recuperar el tiempo perdido al principio de la estancia de la universidad. Al llegar no estaba muy claro el objetivo del proyecto, y perdí tiempo en otros proyectos sin ninguna relación con este.

Estoy muy satisfecho de este motor, es una gran herramienta para el desarrollo de videojuegos, y de las mejores para un desarrollador independiente, tanto por sus posibilidades como el precio (en mi caso gratis). Tanto, que me planteó la posibilidad de crear un video juego, a gran escala, con la ayuda de algunos compañeros, de un género probablemente más comercial, al que estemos acostumbrados a jugar y por tanto nos interese más su desarrollo y en los cuales seamos expertos de la materia. En Hospital Amnesia, sólo he trabajado yo, pero en un juego más comercial, tendrán que participar más personas de distintas disciplinas, como sería un diseñador gráfico o incluso un músico.

Se ha intentado abarcar varios procesos cognitivos, creando tanto un entorno 3d en el que probar algunos más en general, como varios minijuegos en los que se busca estudiar algún proceso más específico. Se iban diseñando y creando a medida que la tesis para optar a doctora de Asma Ben Kheder, los iba requiriendo.

Tanto ella como Claude Frasson, eran expertos en cognición, y guiaron el desarrollo de este juego. También resaltar que era importante tener en cuenta cómo funciona Tobii, el eyetracking que utilizará el laboratorio para las pruebas. El seguimiento de la mirada es complicado y por tanto los puntos de mirada que se buscan seguir debían estar separados, cosa que se tuvo en cuenta en algunos minijuegos.

Las pruebas a pacientes se empezarán en septiembre de 2015, en la Universidad de Montréal, Canadá. Serán realizadas por Asma B. K. y posteriormente analizados los resultados para obtener una conclusión, y obtener respuesta a ¿En qué medida los videojuegos afectan a la cognición?

7.2 Trabajo futuro

El diseño del juego es atractivo en algunas escenas pero en otras deja algo que desear, se buscaba que funcionará y cumpliera su propósito. De todas formas seguiré trabajando en este juego, antes de que empiecen a realizarse las pruebas a pacientes. Realizaré periódicamente subidas a GitHub (github.com/SobeK23).

La ampliación y la mejora de este proyecto es infinita, aunque solo tengo pensado corregir ciertos aspectos visuales, se podrían añadir todo tipo de minijuegos o modificar/añadir escenas, el código se encuentra a disposición de cualquier usuario.

Otra posibilidad sería integrar la página web en el juego, y no tener que salir del juego para posteriormente ingresar los resultados manualmente.

Se podría haber utilizado la Tin Can Api (o Experience API), para realizar la recopilación de datos de los logfiles. Estos podrían ser en JSON, con ayuda de los activity stream, de forma que pudieran enviarse a una base de datos automáticamente. Se podría analizar este tipo de ficheros a partir de la base de datos, en vez de realizarlo manualmente o tener que crear el código necesario para leer este tipo de ficheros log, que usamos en el juego.

En nuestro caso, el juego no podía durar más de 1 hora, puesto que las pruebas que se realizan, necesitan contacto con el cuero cabelludo, y para ello se utiliza un gel que al cabo de una hora deja de funcionar y los resultados pierden su validez. Pero si las pruebas que se realicen en un futuro no necesitan de este gel, se puede ampliar el juego para alargar su duración y realizar otro tipo de pruebas o estas mismas más exhaustivas. Se podría analizar el rendimiento del jugador con el paso del tiempo, así como sus procesos cognitivos durante un largo tiempo y analizar cómo actúa la fatiga.

El estudio de los procesos cognitivos es un campo en estudio todavía, y hay mucho trabajo por delante, sobre todo en el ámbito de los videojuegos. Los cuales cada día toman más importancia en nuestra vida cotidiana, y pueden ser una herramienta muy útil para desarrollar las habilidades cognitivas o simplemente preservar en el caso de gente más anciana.

Este juego es de los primeros en desarrollarse para este campo en específico, ya existen varios serious game en otros campos. Es un buen comienzo para empezar a realizar más estudios y mejorar nuestras habilidades cognitivas.

En un juego es muy importante la historia, que es la que realmente nos sumerge en el juego. En mi caso, la historia es prácticamente inexistente, sería un buen trabajo para el futuro realizar una buena trama, en la que me gustaría que hubiera suspense, para que el jugador quede atrapado en el videojuego. Así llegaríamos a una inmersión total, y el estudio sería aún más detallado.

Y aunque este juego esté enfocado a un estudio específico, se puede reutilizar para otros estudios o simplemente darle otros usos, como pedagógico, modificando algunos mini juegos o añadiendo más. El último juego ya tiene un enfoque pedagógico para estudiantes de medicina, se podría enfocar al campo médico. De esta forma el personal médico podría tanto aprender como repasar los conocimientos de su campo, en un entorno acorde. Gracias a la ayuda de algunos expertos médicos esto podría ser posible. Esto lo realizaron ya (Akl, E. A., Mustafa, R., Slomka, T., Alawneh, A., Vedavalli, A., & Schünemann, 2008) para enseñar a los estudiantes de medicinas algunas guías prácticas. Pero se pueden buscar otros aspectos a enseñar, como diagnósticos médicos.

Glosario

- **Unity 3D:**
Motor y editor gráfico para crear el proyecto.
- **MonoDevelop:**
IDE integrado en Unity 3D para crear y editar scripts en C#, UnityScript o Boo.
- **GameObject:**
Componente básico del motor Unity3D con el que se interactúa mediante el editor.
- **Collider:**
Representación del espacio ocupado por un objeto tridimensional que representa sus límites físicos, es invisible para el jugador.
- **Component:**
Módulo que se añade a un GameObject para agregarle funcionalidad.
- **GUI:**
Interfaz que se muestra en la pantalla del jugador.
- **Cognición:**
Es la facultad para procesar mediante la percepción la información, analizarla y adquirirla a través de lo que llamamos conocimiento.
- **Textura:**
Es una imagen bitmap estándar que se aplica sobre la superficie de una malla.

- **Script:**
Es un archivo de una determinada extensión con código que un programa puede leer. En nuestro caso ficheros C# que interpreta Unity 3d.
- **MonoBehaviour:**
Es la clase base desde la que todos los scripts heredan.
- **GitHub:**
Es una plataforma de desarrollo colaborativo para utilizar el sistema de control de versiones (Git en esta plataforma).

Bibliografía

- Akl, E. A., Mustafa, R., Slomka, T., Alawneh, A., Vedavalli, A., & Schünemann, H. J. (2008). An educational game for teaching clinical practice guidelines to Internal Medicine residents: development, feasibility and acceptability. *BMC Medical Education*, 8(50).
- Anguera, J. a, Boccanfuso, J., Rintoul, J. L., Al-Hashimi, O., Faraji, F., Janowich, J., ... Gazzaley, a. (2013). Video game training enhances cognitive control in older adults. *Nature*, 501(7465), 97–101. <http://doi.org/10.1038/nature12486>
- Barnett, L. (2012). Kids who play interactive games have better motor skills. Retrieved from <http://www.deakin.edu.au/news/media-archives/2012-media-releases-archives/kids-who-play-interactive-video-games-have-better-motor-skills>
- Basak, C., Boot, W. R., Voss, M. W., & Kramer, A. F. (2008). Can training in a real-time strategy video game attenuate cognitive decline in older adults? *Psychology and Aging*, 23(4), 765–777. <http://doi.org/10.1037/a0013494>
- Bulaj, G. (2012). Video Games Help Patients and Health Care Providers. Retrieved from http://unews.utah.edu/news_releases/video-games-help-patients-and-health-care-providers/
- Franceschini, S., Gori, S., Ruffino, M., Viola, S., Molteni, M., & Facoetti, A. (2013). Action video games make dyslexic children read better. *Current Biology*, 23(6), 462–466. <http://doi.org/10.1016/j.cub.2013.01.044>
- Granic, I., Lobel, A., & Engels, R. C. M. E. (2014). The benefits of playing video games. *The American Psychologist*, 69(1), 66–78. <http://doi.org/10.1037/a0034857>

- Green, C. S., & Bavelier, D. (2006). Enumeration versus multiple object tracking: the case of action video game players. *Cognition*, 101(1), 217–245.
<http://doi.org/10.1016/j.cognition.2005.10.004>
- Groundbreaking experiment in virtual reality uses video game to treat pain. (2012). Retrieved from http://rockcenter.nbcnews.com/_news/2012/10/24/14648057-groundbreaking-experiment-in-virtual-reality-uses-video-game-to-treat-pain?lite
- Hasan, Y., Bègue, L., & Bushman, B. J. (2013). Violent Video Games Stress People Out and Make Them More Aggressive. *Aggressive Behavior*, 39(1), 64–70.
<http://doi.org/10.1002/ab.21454>
- Hsu, J. (2010). For the U.S. Military, Video Games Get Serious. Retrieved from <http://www.livescience.com/10022-military-video-games.html>
- Jeon, S. T., Maurer, D., & Lewis, T. L. (2012). The Effect of Video Game Training on the Vision of Adults with Bilateral Deprivation Amblyopia. *Seeing and Perceiving*, 1–28. <http://doi.org/10.1163/18784763-00002391>
- Mejía, C. (2012). Videojuegos y evaluación cognitiva. *Las Tecnologías de La Información En Contextos Educativos: Nuevos Escenarios de Aprendizaje. Sandoval Y Otros (compiladores).*, 223–240.
- Monobehaviour. (n.d.). Retrieved from <http://docs.unity3d.com/ScriptReference/MonoBehaviour.html>
- Shankar, V. (2009). Researchers Explore Mental Health Benefits of Video Games. Retrieved from <http://www.washingtonpost.com/wp-dyn/content/article/2009/08/17/AR2009081702114.html>
- Smith, S. T., & Schoene, D. (2012). The use of exercise-based videogames for training and rehabilitation of physical function in older adults: current practice and guidelines for future research. *Aging Health*. <http://doi.org/10.2217/ahe.12.30>
- Stephen, P. G. (2014). Technology in Academic Medicine: Video Games Take Increasing Role in Medical Education. Retrieved from <https://www.aamc.org/newsroom/reporter/june2014/384790/technology-medical-education.html>
- Unity 3d forum. (n.d.). Retrieved from <http://forum.unity3d.com/>
- Unity answers. (n.d.). Retrieved from <http://answers.unity3d.com/>
- Xian Ying, Kakinami Lisa, Peterson Eric D., Mustian Karen M., and F. I. D. (2014). Will Nintendo “Wii Fit” Get You Fit? An Evaluation of the Energy Expenditure from Active-Play Videogames. *Games for Health Journal*, 3(2).